The Generalized Detail-In-Context Problem [*][†]

T. Alan Keahey

Los Alamos National Laboratory, MS B287
Los Alamos, NM 87545
keahey@lanl.gov

Abstract

This paper describes a general formulation of the "detail-in-context" problem, which is a central issue of fundamental importance to a wide variety of nonlinear magnification systems. A number of tools are described for dealing with this problem effectively. These tools can be applied to any continuous nonlinear magnification system, and are not tied to specific implementation features of the system that produced the original transformation. Of particular interest is the development of "seamless multi-level views", which allow multiple global views of an information space (each having different information content) to be integrated into a single view without discontinuity.

1. Introduction

Many approaches have been described in the literature for stretching and distorting spaces to produce effective visualizations of data. Such techniques have been called *polyfocal projection*[8], *fisheye views*[7], *distortion-oriented presentation*[15], *focus + context*[14] and many other terms [10]. In [12] we introduced the term *nonlinear magnification* to describe the effects common to all of these systems. The basic characteristics of nonlinear magnification are non-occluding in-place magnification which preserves a view of the global context. In this paper we will define the *detail-in-context problem* as a general issue of significance to all nonlinear magnification systems, and then describe a collection of methods for dealing with the problem. Both the statement of the problem, and the methods for dealing with it are very general-purpose in nature, and can be readily applied to existing nonlinear magnification systems.

In overview: after first defining the problem, we will provide a brief review of the nonlinear magnification fields which are of central importance to the techniques described later. Following this we will examine specific methods for addressing the detail-in-context problem, concentrating first on the case for discrete objects, and then considering the task of seamlessly integrating different global views of the information space. We finish with a brief discussion on consistent visual cues for magnification, followed by related work and conclusions.

1.1. The Detail-In-Context Problem

The detail-in-context problem for visualization with nonlinear magnification systems can be stated briefly as:

> How can we effectively utilize the additional space made available by *any* nonlinear magnification transformation to enhance the visualization of the data or objects located within that space?

Figure 1 shows a diagram of the problem. Detail can be seen as an additional axis that is orthogonal to the transformational axes. The interpretation of this detail axis is highly task-dependent. It can refer to something as simple as object size, or as complex as semantic levels of information content. One goal of this paper is to provide a single, unifying method for defining the detail levels implicit in a nonlinear magnification system, so that more sophisticated treatment of the detail axis can then be based on those defined values.
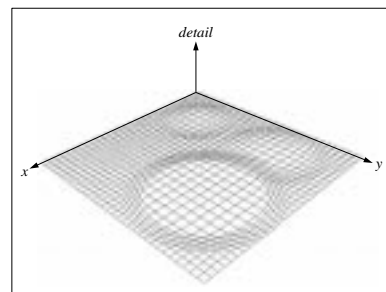

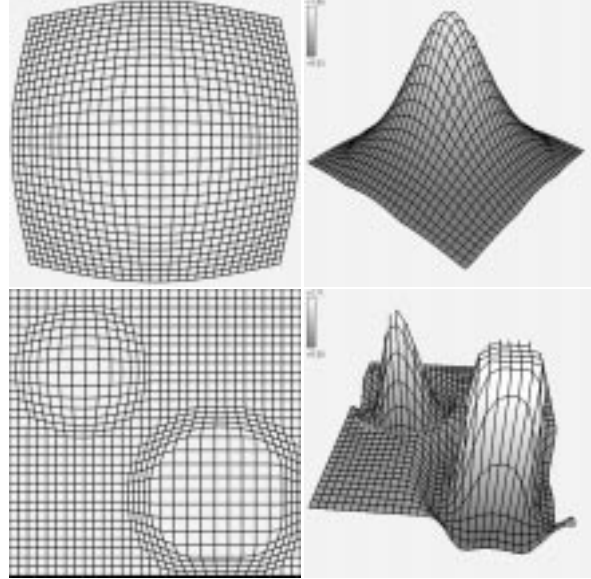**Figure 1. The Detail-In-Context Problem**

The formulation of this problem is similar to a number of approaches already taken in the literature. The Perspective Wall [17] was an effort to provide "detail and context smoothly integrated". Researchers at Xerox PARC using a 2D hyperbolic transformation for display of graphs referred to their technique as a "focus+context" technique [14]. Similar terminology is scattered through much of the literature since that time. Our definition of the problem has significant differences from these specific approaches however, which we will now outline.

Many focus+context techniques are designed to create "focus" by enlarging spaces, and reduced "context" by compressing the surrounding spaces. This addresses only half of the detail-in-context problem as it has been defined here; it creates the space needed for additional detail, but does not by itself provide a means for placing more detail within that space. Although many of these systems also provide enhanced detail within regions of magnification [22, 19, 23, 1, 14, 18]; the techniques that we will describe in this paper are of a more general nature, these methods effectively synchronize detail functions with *any* continuous nonlinear magnification transformation. Although most of the transformations in this paper were generated using the nonlinear magnification transformations described in [12], the techniques which we will explore here are independent of the actual mechanism used to produce the original nonlinear transformation.

1.2. Nonlinear Magnification Fields

Leung and Apperley [15] first established the mathematical relationship between magnification and transformation functions for nonlinear magnification or "distortion-oriented" systems. For the one-dimensional case they define the magnification function as the derivative of the transformation function. This relationship was extended to higher-dimensional nonlinear transformations in [13], resulting in the *nonlinear magnification field* (described in greater detail in [9]). Methods were provided for converting back and forth between 2D nonlinear transformation functions and their associated magnification functions. It was shown that every continuous, order-preserving, nonlinear transformation has an *implicit magnification field* which is a field of scalar values reflecting the magnificational effect of that transformation. Implicit magnification fields are very inexpensive to compute, provide a consistent measurement of the effect of transformations, and can be computed on any continuous nonlinear magnification transformation, regardless of what type of system was used to produce that transformation (an example shown in [13] illustrates the implicit magnification field of the Perspective Wall [17] system). These magnification fields play a central role in the methods described in this paper; their implementation-independent nature provides a general-purpose method for quantifying

the effects of specific nonlinear magnification transformations, thus providing a rigorous mathematical measure on which to synchronize our detail-rendering methods. Figure 2 shows two examples of nonlinear transformations along with their associated implicit magnification fields.

Figure 2. Transformations and their Implicit Magnification Fields

2. Putting Detail In Context

In this section we will examine the problem of how to provide suitable detail within magnified regions of a nonlinearly transformed domain. First we will consider the case where our information is composed of discrete objects within the coordinate space, and then we will consider the problem of seamlessly integrating different global views of the information space (each view having a different information content). Throughout the section we will use an "interactive travel atlas" as an application to demonstrate the concepts involved. As regions of the atlas are expanded with nonlinear magnification by a user, points of interest within those regions can be displayed accordingly. For this specific example we will use an atlas of Scotland, with major points of interest (castles, whisky distilleries, etc.) as the discrete data objects within the map. Each atlas example is illustrated both in the paper and in Colour Plate A. These examples make extensive use of texture mapping techniques [3] to place the images on the screen.

2.1. Discrete Objects

The first general detail-in-context task that we will examine involves rendering discrete objects within a nonlinearly
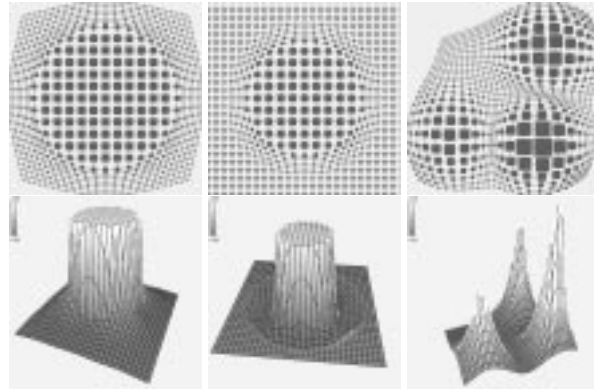
transformed space. The problem is to determine how to render these discrete objects in a manner that is consistent with the underlying spatial transformation. There are several ways in which this can be approached, ranging from simple object-size calculations to "embedding" objects within the underlying space. We will now examine these methods individually, and also show different ways in which they can be combined.

2.1.1. Object Size

The simplest method for increasing detail of objects involves only increasing their size. This method is commonly used for single-foci systems such as [22], where object size can be based on simple Euclidean proximity to the center of magnification. However the task becomes more difficult when complex transformations with multiple foci and/or constrained domains are used; simple Euclidean distance is no longer effective as a measure on which to base object size in such cases. A recent article [6] describes the separation of transformation ("displacement") and magnification (a conceptual distinction describing node size) functions, however the authors do not address the issue of how to ensure that these functions are reasonably synchronized. All of the examples shown in that work involve either a very simple single-focus transformation function or else a very simple magnification function; for such cases there is no complex interaction of transformation and magnification to account for, and thus simple proximity-based approaches can be used for determining detail.

The implicit magnification fields developed in [13] are very well suited for the task of synchronizing transformation and magnification functions when complex transformations are involved. By computing the implicit magnification field for the transformation we can find the magnification for any object within the transformation domain, and render the object with a size proportional to that magnification. This method is general-purpose in nature, and does not require any special knowledge about foci-location or other facts that are internal to the specific transformation technique used to produce the transformation. In addition, since the implicit magnification field is C^0 continuous and well defined over the entire domain it does not leave any gaps where the magnification is undefined, as is the case for some of the other approaches for graph visualization that only define magnification locally at the nodes of the graph [19]. Figure 3 shows several examples of how object size can be coupled effectively with implicit magnification values. [1]

Figure 3. Synchronizing Object Size and Implicit Magnification

Figure 4 shows how object-size rendering might work for our interactive atlas; each point of interest is rendered as an image which is uniformly magnified proportionally to the implicit magnification of the transformation. Although this example illustrates effective synchronization of detail and transformation functions, it also shows how object size alone is not always sufficient to guarantee that the objects do not overlap each other. In this example we sort the images by the implicit magnification level of the transformation, so that the highest-magnified image will always be completely visible. This sorting can be performed analytically, or on a per-pixel basis using z-buffer rendering. A more sophisticated approach to this problem which uses embedded objects will also be described in Section 2.1.3.

Figure 4. Interactive Atlas with Variable Object Sizing

2.1.2. Level of Detail

We can extend the object-size methods by incorporating *level of detail* (LOD) for the rendering of the objects. Level of detail is a common technique in 3D graphics. It is typically used to suppress details in the polygonal representation of objects when the object is far away from the

---

[1]Uniform scaling is used in these examples, although non-uniform aspect ratios are also possible. This was illustrated for simple cases in [6], and can be implemented for complex cases via the nonlinear magnification vectors described in [9].

view-point, since the detail would not be visible at the pixel level anyway.

The LOD notion can be generalized to tasks other than polygonal simplification however, and can incorporate concepts such as semantic levels of detail. Using our interactive atlas example, we might want to represent each castle in the atlas with three levels of detail. At level 0 we can represent the castle by a picture of it, at level 1 we use an iconic representation of a castle (which is shared by all castles), and at level 2 we simply represent the castle by a coloured square. Figure 5 shows a schematic representation of these levels of detail.

Figure 5. Castle Levels of Detail

We can easily drive the LOD rendering of objects simply by making the level of detail proportional to the implicit magnification produced by the transformation. Figure 6 shows an example of our interactive atlas using both LOD and object size based on implicit magnification. This is but one simple example of how LOD rendering can be incorporated into nonlinearly magnified spaces, many other approaches are possible; for example the Pad++ [2] WWW navigation system uses page thumbnails with a LOD function so that the node at the focus of the transformation becomes an actual web page which the user can interact with, other systems for visualization of graph structures expand and collapse subgraphs as their root nodes are magnified or demagnified [19].

At this point we are still left with the problem that simple linear scaling of objects proportionally to the implicit magnification of the transformation can result in overlapping objects. In addition, this method leads to the perception of the objects as "floating" above the transformed space. Therefore while this method binds the dimensions of detail and context effectively, it does not reflect more complex aspects of the transformation itself within the objects.

2.1.3. Embedded Objects

An alternative to the simple object-size approach is to *embed* the objects within the transformed coordinate space. This goes beyond simply placing the centers of objects appropriately within the transformed space, and involves mapping the *boundaries* of the objects to the transformed spatial coordinates. Embedding objects in this way produces

Figure 6. Interactive Atlas with LOD and Variable Object Sizing

what could be called a *coherent information space*, where the objects obey the same "transformational physics" as the underlying space. The result is a visualization that has a more tangible aspect to it; the magnification produced by the transformation can now be perceived consistently on three different levels: on the underlying space, between objects, and within individual objects. Prominent examples of this type of embedded object are found in the Perspective Wall [17] and the Document Lens [21]. Figure 7 shows an example of using embedded objects for our interactive atlas.

Figure 7. Interactive Atlas with Embedded Objects

There are a number of problems with embedded objects however. The first problem is that magnification level for the objects is *directly* proportional to the implicit magnification of the underlying transformation, and is therefore not as flexible and responsive as using object size alone, where possibilities such as making object size proportional to the square of the implicit magnification allow for a greater dynamic scaling range for the objects. Another problem with embedded objects is that layout of the objects within the

original untransformed space becomes more of a challenge if we want to ensure that the objects do not overlap at any point along their boundaries, since we need to ensure that the objects do not overlap in the original untransformed space. This difficulty is by no means insurmountable, but does produce constraints on the size and location of objects within the original coordinate space. A final problem with embedded objects is that they may introduce distortion within the objects being magnified. This problem can be dealt with by using transformations with linear regions of magnification to provide uniform scaling within the magnified regions, as shown in Figure 8.

Figure 8. Interactive Atlas with Embedded Objects and Linear Magnification (compare the linear scaling of the castle in focus with the distortion of the same object in Figure 7)

2.1.4. Embedded Objects with Size and Level of Detail

We can combine all of the previous techniques within a single visualization system so that we get the advantages of each technique. We linearly scale the objects in the original layout based on their implicit magnification value in the transformed space (clamping the scale factor so that they do not scale beyond the boundaries defined in the initial layout). Then LOD filtering can be applied, and finally the boundaries of the scaled, filtered object are mapped to the transformed coordinates to embed the object within the transformed space. Figure 9 shows an example of this with our interactive atlas. Note that the maximum size of the objects is still bounded with this system, we can remove this restriction by allowing the objects to scale beyond the initial layout boundaries, using sorting based on magnification (either on a per-object or per-pixel basis) to manage the overlapping objects. This damages the coherency of the information space somewhat, but the extra benefits of allowing larger objects may make this worthwhile.
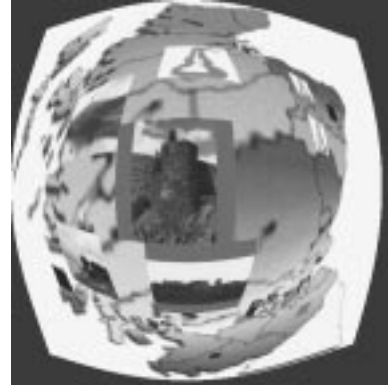

Figure 9. Interactive Atlas with Variable Size, LOD and Embedded Objects

2.2. Seamless Multi-Level Views

Another level on which the detail-in-context problem can be addressed involves integrating different global "views" of the information space. The term "view" in this context does not refer to a different physical viewpoint, but rather a different visual representation of the information space. In the specific examples described here, each view is represented by a discrete image. This does not represent a fundamental limitation on the types of views or data that can be used, since any data can always be rendered to an off-screen buffer and then used as an image (although this may introduce implementation considerations).

Since we will be dealing with views as images, it is illustrative to look first at the example of nonlinear magnification of a single image. Figure 10 shows a simple 8×8 checkerboard image alongside a nonlinearly magnified version of the image. Because the data content is static between normal and magnified versions, no additional information is obtained through the nonlinear magnification, and the only thing that changes is the size (and shape) of the original 8×8 squares. Some filtering techniques which are commonly used in texture mapping may also allow interpolated values between the data squares, however this does not provide any new information to the user, but rather just a smoother transition between normal and magnified squares. For cases where the original image is larger than the available screen pixels, similar filtering techniques can be used to downsample the image so that the low frequency content of the entire image is still visible. Magnification of these downsampled regions may reveal a clearer view of the actual image pixels, however the underlying image content does not change, only the sampling frequency of that image. Nonlinear magnification of individual images was illustrated in [4]; details of nonlinear image magnification and the differences between discrete and continuous domains were described extensively in [11], and later mentioned in [5].
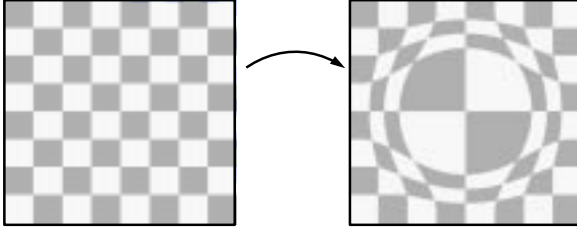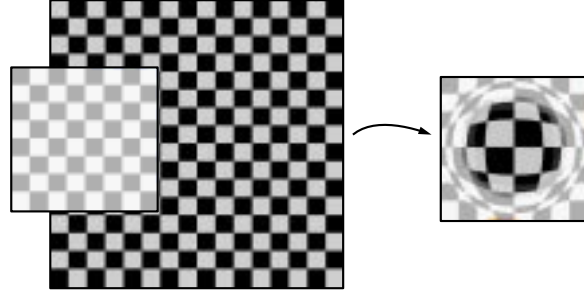
Figure 10. Single Image Magnification

Figure 11. Multi-Level Image Magnification

We can extend this idea of image magnification to account for multiple levels of images (i.e. multiple views), where each view can now represent distinct semantic or graphical representations of the overall information space (this idea was first mentioned in [11]). An example application where this might be useful involves combining state, county and city maps within a single magnified view. At the top level the user is looking at a state map; as the user magnifies some region of the map, the county map is "pulled in" to the magnified area to provide that detail within the state map. Further magnification would also pull in additional detail from the city map.

We will first examine a simple example of multi-level image magnification to illustrate the issues that are involved. Consider two independent views of an information space, the first view is an 8×8 grid, and the second view is a 16×16 grid (the grid sizes here are deliberately chosen to illustrate the effectiveness of integrating the two views, in practice any other grid sizes could be used). Conceptually, we can think of this process as looking straight-on at the centers of the images, with the 8×8 image in front of the 16×16 image, and filling the entire window. As we apply nonlinear magnification, we effectively "punch a hole" through the 8×8 grid and pull in the view of the 16×16 grid so that the two views are seamlessly integrated. Figure 11 shows an example of this operation; notice how each square of the 8×8 is perfectly aligned with the corresponding 4 squares from the 16×16 grid, and note also how the two images are blended together around the region of magnification to provide a smooth transition between images. This technique differs significantly from single image magnification in that we are now dynamically incorporating additional detail within the context provided by our nonlinear magnification transformation.

Figure 12 shows an example of this using two different maps from the Xerox PARC Map Server [20]. Two views of the California Bay Area are provided, with the larger map showing more detail (roads, railway tracks etc.) than the smaller one. As the smaller, simple view is magnified, additional information is pulled in from the detailed view. Note

that the integration of the two views is seamless, and that all of the map lines are perfectly aligned at the intersection of the simple and detailed views. Colour Plate B shows our two examples of seamless multi-level views.
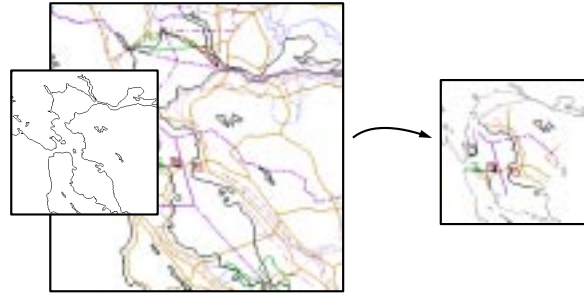

Figure 12. Multi-Level Map Magnification

The implementation of multi-level image magnification can be greatly facilitated by the use of MIP-mapping [25], which is a common technique within the graphics community for dealing with texture mapping, and is supported by hardware acceleration on most workstations and PCs with hardware graphics capabilities. MIP-mapping is a method for storing different resolution versions of a texture map, so that the most appropriate resolution level can be used for the patch that the texture is being applied to. For example, an $n \times n$ texture will be stored at the original resolution at level 0, along with a filtered $n/2 \times n/2$ version at level 1, and a $n/4 \times n/4$ at level 2 and so on down to a 1×1 version at level $\log_2 n$. Figure 13 shows a schematic representation of this for a single channel of an RGB texture.

We can bypass the normal filtering construction of MIP-map levels and load any image into the different levels of the MIP-map, as long as the image has the same number of pixel rows and columns as are required for that level. Using our previous example, we can load the 256×256 pixel image of the 8×8 grid into level 1, and the 512×512 pixel image of the 16×16 grid into level 0. If we size our view-port to the same number of pixels as the level 1
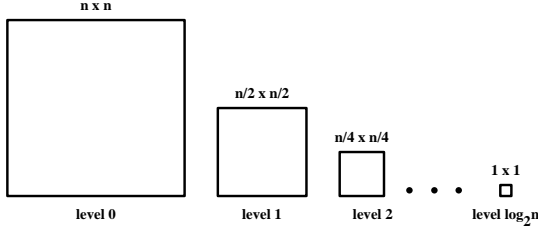
Figure 13. MIP-Mapping for a Single Channel

image (256×256), we will see only that image; as we magnify portions of the level 1 image, the level 0 image will be pulled into the context of the level 1 image. Although this method works very efficiently on hardware accelerated machines, some hardware implementations also place limits on the size of images that are allowed. This can be a limitation on the maximum size of image (typically somewhere between 512×512 and 2048×2048 pixels), or on the scale factor between levels (some graphics library implementations restrict this to a factor of 2). Although workarounds can usually be found for these constraints, an area for further research is to develop a more general formalism for describing multi-level image magnification outside of the constrained hardware-accelerated environment.

3. Consistent Visual Cues

A final issue involved with the detail-in-context problem is the need to provide consistent visual cues to the user as to what regions are being magnified or demagnified by a given transformation. This need was addressed in an implementation-specific manner for the 3DPS system [4], through the combination of an additional NURB surface and a computationally expensive lighting model to produce shading of regions of distortion. More generally, suitable shading for any given transformation can also be produced in an *implementation-independent* fashion through the use of implicit magnification fields. These provide a consistent quantification for the degree of magnification implicit in a given transformation, making our task simply to render the information in a way that reflects this quantification (the 3DPS system does not provide a mechanism for quantifying the effects of a given transformation, and has an inconsistent relationship between elevation and magnification [13]). We have already seen examples in Section 1.2 where the implicit magnification values are mapped into a 1D colour ramp to provide consistent visual cues for a single surface; the situation is somewhat more complicated for textured or multiple surfaces however. One possibility is to use multi-pass rendering and modulate all of the surfaces with the appropriate colour ramp values during one of the passes. Another simpler method involves mapping the surfaces onto a composite mesh (defined in [9], each node in the composite

mesh has the $\{x, y\}$ coordinates of the transformation grid, and the z value of the implicit magnification mesh of the transformation). The mesh is then viewed from above with an orthographic transformation, and fog is used to gradually fade out the regions of lower magnification. The fog colour can be set to any RGB values, depending on what colour works best for a particular application. This technique is illustrated for both a single textured surface and a multi-level view in Figure 14 and Colour Plate C. The fog-based approach to providing visual cues has the benefit of allowing for very simple implementation (most current graphics libraries provide ready support for this), and is also very inexpensive computationally (having hardware support on many platforms).

Figure 14. Using Fog to Indicate Magnification

4. Related Work

Lieberman [16] takes a different approach to integrating global views of an information space. The primary technique used there relies on overlapping layers of varying translucency, so that the different global views are visible simultaneously. In contrast, the seamless multi-level views described here do not introduce the clutter of overlapping images; in general each pixel will be associated with a single view, except at the transition zone between two different views, where it will be a blended value from between those two views. Another fundamental difference is that Lieberman's method leaves sharp spatial discontinuities between the levels of information. Although these discontinuities help to facilitate large degrees of magnification, they also place the method outside the scope of the traditional nonlinear magnification techniques which seek a smoother transition between the magnified and compressed regions in the spatial plane. In contrast, the seamless multi-level views presented here provide smooth transitions that are free of discontinuities.

Magic Lens [24] filters provide many different methods for changing the visual representation of information as the filters pass over the workspace. Filters are available for increasing or decreasing detail, as well as for altering semantic representation and other effects. A key difference between these filters and the methods described here is that

the filters are implemented as distinct objects with discrete boundaries, and do not create the nonlinear spatial transformations which expand and compress the space to allow more or less detail. The tools described in this paper differ in that their effects are all defined by the intrinsic properties of a given nonlinear transformation, in effect allowing the nonlinear transformation to drive the filtering process in a well-synchronized fashion.

5. Conclusions

In this paper we have defined a general statement of the "detail-in-context" problem common to many nonlinear magnification systems. By defining the general case for this problem, we allow for the analysis and construction of techniques for dealing with the problem that are not tied to the specific implementation details of the system that is producing the original nonlinear magnification transformation.

We have explored several different techniques for handling the detail-in-context problem. Some of these techniques are primarily a generalization of methods that have been described elsewhere in the literature, whereas other techniques (such as seamless multi-level views) offer a completely novel functionality. All of these methods are based on our low-level implicit magnification field method for defining well-synchronized detail values for a given nonlinear transformation. Through combinations of the techniques developed here, we can define general-purpose methods for enriching the visualization of information spaces that have been transformed via the nonlinear magnification paradigm.

References

[1] L. Bartram, A. Ho, J. Dill, and F. Henigman. The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Nov. 1995.

[2] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas. Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages and Computing*, pages 3–31, 1996.

[3] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–546, Oct. 1976.

[4] M. Carpendale, D. Cowperthwaite, and D. Fracchia. 3D pliable surfaces: For the effective presentation of visual information. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 217–226, 1995.

[5] M. Carpendale, D. Cowperthwaite, and F. Fracchia. Multiscale viewing. In *SIGGRAPH Visual Proceedings*, page 149, Aug. 1996. Technical Sketch.

[6] M. Carpendale, D. Cowperthwaite, and F. Fracchia. Extending distortion viewing from 2D to 3D. *Computer Graphics and Applications*, 17(4):42–51, July 1997.

[7] G. W. Furnas. Generalized fisheye views. *Human Factors in Computing Systems, CHI '86*, pages 16–23, Apr. 1986.

[8] N. Kadmon and E. Shlomi. A polyfocal projection for statistical surfaces. *The Cartographic Journal*, 15(1):36–41, June 1978.

[9] T. A. Keahey. *Nonlinear Magnification*. PhD thesis, Department of Computer Science, Indiana University, Dec. 1997.

[10] T. A. Keahey. the Nonlinear Magnification Home Page. A WWW resource devoted to all aspects of nonlinear magnification available at: www.cs.indiana.edu/hyplan/tkeahey/research/nlm/nlm.html, Jan. 1997.

[11] T. A. Keahey and E. L. Robertson. Non-linear image magnification. Technical Report 460, Department of Computer Science, Indiana University, Apr. 1996.

[12] T. A. Keahey and E. L. Robertson. Techniques for non-linear magnification transformations. In *Proceedings of the IEEE Symposium on Information Visualization, IEEE Visualization*, pages 38–45, Oct. 1996.

[13] T. A. Keahey and E. L. Robertson. Nonlinear magnification fields. In *Proceedings of the IEEE Symposium on Information Visualization, IEEE Visualization*, Oct. 1997.

[14] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the ACM Conference on Computer Human Interaction*, 1995.

[15] Y. Leung and M. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.

[16] H. Lieberman. Powers of ten thousand. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 15–16, Nov. 1994. Demo.

[17] J. Mackinlay, G. Robertson, and S. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the ACM Conference on Computer Human Interaction*, pages 173–179, 1991.

[18] T. Munzner. H3: Laying out large directed graphs in 3D hyperbolic space. In *Proceedings of the IEEE Symposium on Information Visualization, IEEE Visualization*, Oct. 1997.

[19] E. G. Noik. Exploring large hyperdocuments: Fisheye views of nested networks. In *Proceedings of the ACM Conference on Hypertext*, pages 192–205, Nov. 1993.

[20] S. Putz. Xerox PARC map viewer, 1993. www.parc.xerox.com/istl/projects/mapdocs/.

[21] G. Robertson and J. D. Mackinlay. The document lens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 101–108, 1993.

[22] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. In *Proceedings of the ACM Conference on Computer Human Interaction*, May 1992.

[23] M. Sarkar, S. S. Snibbe, O. Tversky, and S. P. Reiss. Stretching the rubber sheet: A metaphor for visualizing large layouts on small screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1993.

[24] M. C. Stone, K. Fishkin, and E. A. Bier. The movable filter as a user interface tool. In *Proceedings of the ACM Conference on Computer Human Interaction*, 1994.

[25] L. Williams. Pyramidal parametrics. In *SIGGRAPH*, 1983.

Colour Plate A: Interactive Travel Atlas (left to right, top to bottom: size only, size with LOD, embedded, embedded with linear, embedded with size and LOD)

Colour Plate B: Seamless Multi–Level Views

Colour Plate C: Fog Cues